МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет им. Н. И. Лобачевского»

Радиофизический факультет

Направление 02.03.02 «Фундаментальная информатика и информационные технологии»
Профиль «Информационные системы и технологии»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Глубокие нейронные сети для распознования формул на изображениях

«К защите допущен»:

Зав. кафедрой статистической радиофизики
и мобильных систем связи,
профессор, д.ф.-м.н. Мальцев А.А.

Научный руководитель,
профессор, д.ф.-м.н. Еськин В.А.

Рецензент,
доцент, к.ф.-м.н. Еськин В.А.

Консультант по технике безопасности
доцент, к.ф.-м.н. Клемина А.В.

Студент 4-го курса Новиков Е.А.

Нижний Новгород
2022

Содержание

Введение

На данном этапе развития технологий в современном мире всё чаще появляется необходимость автоматизированного чтения текста с изображений, фото или видео. Под текстом может подразумеваться абсолютно всё: рукописный текст, формулы, требующие переноски из старых учебников и книг в сеть или быстрый перевод текста с бумажных носителей. В любом случае, такие технологии сильно облегают жизнь людей.

Сейчас такую возможность нам предоставляют нейронные сети, которые, по сути, имитируют некоторые аспекты умственной деятельности человека, так как нейронная сеть – это модель, математически созданная на основе биологических нейронных сетей и их функционирования. Первая попытка создания нейронной сети принадлежит Уоррену Мак-Каллоку и Уолтеру Питтсу, которые формируют понятие нейронной сети [1]. А через несколько лет Дональд Хебб предлагает первый алгоритм обучения.

Интерес к нейронным сетям обусловлен их успешным применением в самых разных областях – медицина, бизнес, геология, физика. Их практикуют везде, где нужно находить решения задач управления, классификации или прогнозирования. Так Бернард Уидроу и его студент Хофф создали Адалин, использовавшийся для задач предсказания и адаптивного управления [2]. В 2007 году Джеффри Хитоном в университете Торонто созданы алгоритмы глубокого обучения многослойных нейронных сетей. Для этого была использована ограниченная машина Больцмана[3]. Для обучения должно использоваться большое количество образов, которые могут быть распознанны. После обучения на выходе имеется быстро работающая программа с возможность. решения конкретных задач.

Целью работы является обучение нейросети, основанной на GRU ячейке и нейросети, основанной на модели "Трансформер с последующим сравнением их результатов.

Актуальность данной работы состоит в сравнении двух нейронных сетей, разработанных на двух различных моделях. Данное решение принято исходя из того, чтоб подсчитать, с каким успехом развиваются нейронные сети и с какой скоростью они будут обучаться, имея одинаковый набор данных.

Данная работа состоит из двух глав. В первой главе рассматривается эксперимент с распознаванием формул на изображениях глубокой нейронной сетью, основанной на GRU ячейке. Во второй исследуется распознавания формул на изображениях глубокой нейронной сетью, основанной на модели "Трансформер".

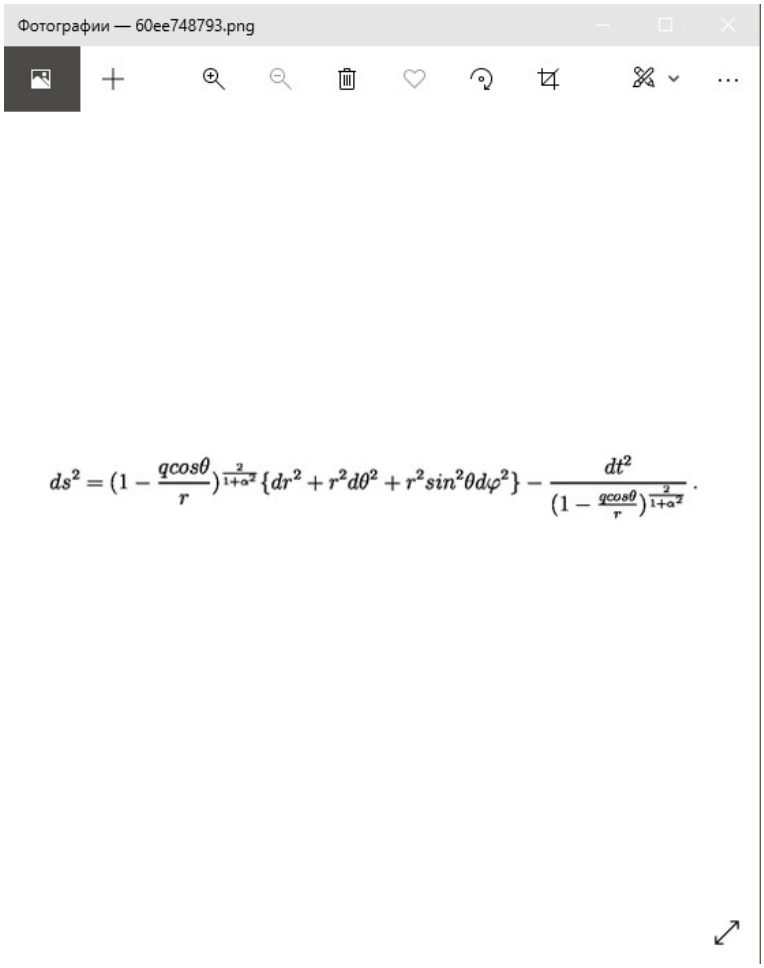
1

Распознавания формул на изображениях глубокой нейронной сетью, основанной на GRU ячейке

1.1. Описание набора данных и подготовки набора данных

В качестве набора данных используются изображения с формулами, взятых с гарвардского проекта[4]. Набор состоит из двух пакетов "generateset"и ”hugedataset”. Первый представляет собой изображения с формулами в количестве 1700 штук, второй 104000 изображений. "generatedset"будет использоваться только для первой итерации каждой нейронной сети, для последующих итераций будет использован ”hugedataset”. Изначально мы имеем изображения формул на листе А4.

Для оптимизации работы нейронной сети, предварительно, эти изображения обрезаются.



Создаётся отдельный файл, где каждая из формул прописана в печатном виде и имеет свой номер. После чего, формулы нормализуются. Для наибольшего успеха обучения, из пакета данных формулы исключаются те, что имеют большое количество токенов и грамматические ошибки.

1.2. Описание модели

Данная сеть представляет собой свёрточную нейронную сеть с несколькими стандартными нейронными компонентами из области зрения и обработки естественного языка. Сначала он извлекает объекты изображений с помощью свёрточной сети (CNN) и упорядочивает объекты в сетке, затем каждая строка кодируется с помощью рекурентной сети (RNN). После используется декодер с механизмом внимания. Визуальные признаки изображений извлекаются с помощью многослойной нейронной сети.Формально рекуррентная нейронная сеть представляет собой параметризированную функцию RNN, которая рекурсивно отображает входной вектор и скрытое

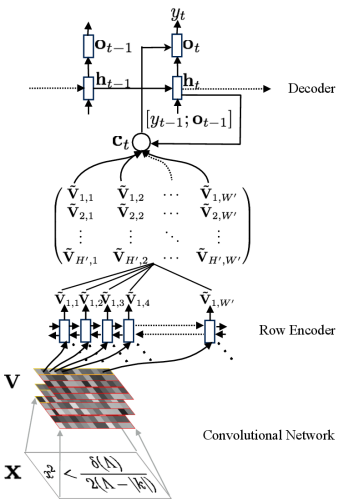


Рис. 1.1. Схема свёрточной сети, использованной в обучении.

состояние в новое скрытое состояние.

В момент времени t скрытое состояние обновляется с помощью ввода $v_t : h_t = RNN(h_t - 1, v_t; \theta)$, где h_0 - начальное состояние.

В этой модели новая сетка объектов $\widetilde{V}$ создаётся, при запуске RNN по каждой строке этого ввода. Рекурсивно для всех строк $h \in (1, ..., H')$ и столбцов $w \in (1, ..., W')$, новые объекты определяются как $\widetilde{V}_{h,w} = RNN(\widetilde{V}_{h,w-1}, \widetilde{V}_{h,w})$.

Для того, чтобы захватить информацию о последовательном порядке в вертикальном направлении, используется обучаемое начальное скрытое состояние $\widetilde{V}_{h,0}$ для каждой строки.

Далее декодером генерируются марки разметки y_t на основе последовательности сетки аннотаций $\widetilde{V}$. Вектор h_t используется для суммирования истории декодирования: $h_t = RNN(h_{t-1}, [y_{t-1}; o_{t-1}])$[5].

1.3. Описание процесса обучения и оценки точности модели

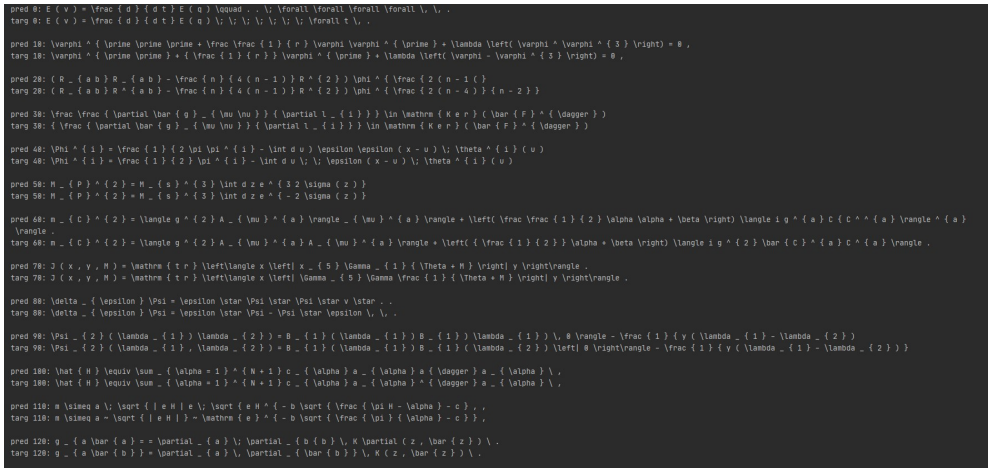
В самом начале нужно подготовить среду обучения нейронной сети. Сам процесс обучения состоит из 11 итераций(заданий), для каждой из которых введены свои критерии обучения. К примеру для первой задаются такие параметры:

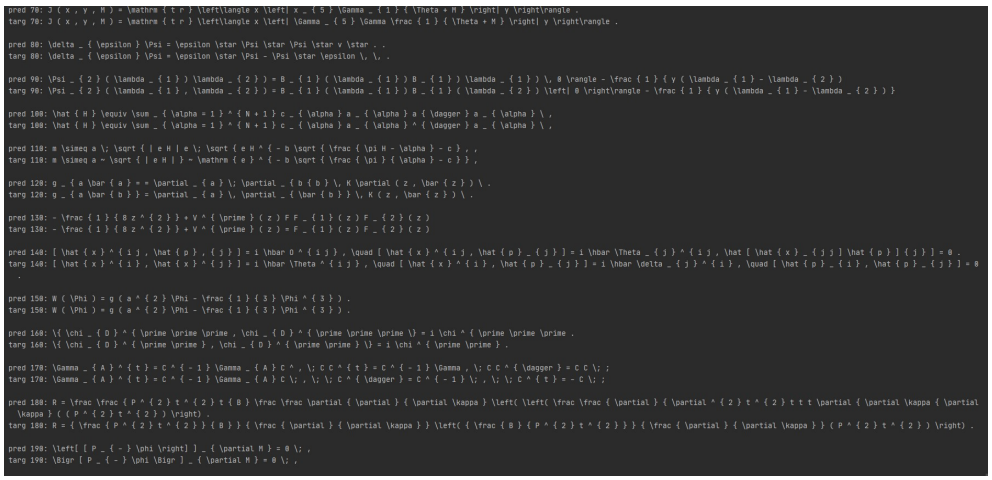
1. Learning rate = 0.001

2. Number of epochs = 100
3. Batch size = 20
4. ENC_DROPOUT = 0.5
5. DEC_DROPOUT= 0.5
6. DownsampleImage = False
7. Height of image = 60
8. Width of image = 360

Так как это самая первая итерация, тут нейронная сеть обучается 100 эпох на минимальном наборе изображений (1200 элементов). Но всё дальнейшее обучение будет проходить на большом "hugedataset"наборе изображений (104 000 элементов).

Единственным результатом обучения нейронной сети, который мы можем увидеть является проверочная часть в конце каждой эпохи и выглядит она так:





Это результат проверочной работы нейросети после 100-ой эпохи обучения на наборе изображений "generatedset".

После прохождения одной полной итерации, мы получаем оценку работы нейронной сети, в частности расчёт её ошибки и BLEU (Bi-Lingual Evaluation Understudy) значение, которое показывает насколько распознанная нейронной сетью формула совпадает с правильной фомулой.

Validate average loss: 8.536643981933594

BLEU score = 12.42

По мере прохождения обучения параметры нейронной сети будут изменяться следующим образом. После первой итерации будет изменён пакет данных с "generatedset"на "hugedataset". Далее для третьей итерации будет изменено значение "Learning rate":

Learning rate = 0.01

Для четвёртой итерации снова будет изменён параметр "Learning rate":

Learning rate = 0.0001

Для пятой обновлено значение "Batch size":

Batch size = 16

Перед запуском шестой итерации изменяем значение "DownsampleImage"с "False"на "True оно останется в этом положении до окончания обучения. Для запуска седьмой, поменяем сразу три параметра:

1. Learning rate = 0.00001
2. ENC_DROPOUT = 0.1
3. DEC_DROPOUT = 0.1

Для восьмой выставим исходное значение переменных "DropOut":

1. ENC_DROPOUT = 0.5
2. DEC_DROPOUT = 0.5

Для девятой сделаем такие изменения и переведём изображения в новый формат:

1. Learning rate = 0.0001
2. Batch size = 6
3. Height of image = 128
4. Width of image = 512

Для десятой внесём такие значения:

1. Learning rate = 0.00001
2. ENC_DROPOUT = 0.1
3. DEC_DROPOUT = 0.1

И, для завершающей обучение, одиннадцатой итерации изменяем эти параметры:

1. Batch size = 3
2. ENC_DROPOUT = 0.0
3. DEC_DROPOUT = 0.0

1.4. Результаты численных экспериментов

Итоги первой итерации:
Validate average loss: 8.536643981933594
BLEU score = 12.42

Итоги второй итерации:
Validate average loss: 2.291751463846753
BLEU score = 23.42

Итоги третьей итерации:
Validate average loss: 3.773274381954685
BLEU score = 2.87

Итоги четвёртой итерации:
Validate average loss: 4.963756845783645
BLEU score = 10.19

Итоги пятой итерации:
Validate average loss: 4.795395610563912
BLEU score = 14.93

Итоги шестой итерации:
Validate average loss: 8.862708091735844
BLEU score = 16.16

Итоги седьмой итерации:
Validate average loss: 4.913547956497258
BLEU score = 14.04

Итоги восьмой итерации:
Validate average loss: 5.368844568714796
BLEU score = 9.18

Итоги девятой итерации:
Validate average loss: 5.017368428675382
BLEU score = 5.70

Итоги десятой итерации:
Validate average loss: 5.047398713594534
BLEU score = 8.56

Итоги одиннадцатой итерации:
Validate average loss: 3.983458673284735
BLEU score = 11.46

Анализируя полученные результаты, стоит отметить, что при смене пакета изображений на "hugedataset мы наблюдаем сильное улучшение результатов. Но, для первой и второй итераций параметры были неизменны. После прохождения 3 итерации можно заметить сильный спад результатов обучения, что и связанно с началом изменения критерий обучения. Пятая, шестая и

седьмая итерации, по результатам обучения, примерно одинаковы. Для девятой итерации был введён новый формат изображений из-за чего качество распознавания нейронной сетью сильно снижается, но в конечном итоге мы получаем, судя по нашим результатам, среднее значение для BLEU score и Validate average loss.

2

Распознавания формул на изображениях глубокой нейронной сетью, основанной на трансформере

2.1. Описание модели

Нейронная сеть с моделью "Трансформер"так же как и первая модель является свёрточной и состоит из слоёв. Отличие её в том, что для отпимизации её скорости и обучения , она оснащена "механизмом внимания". Вместо того, чтобы полученная в процессе обучения информация переходила из одного слоя в другой, используется механизм, который принимает решение какой элемент входной последовательности имеет важное значение для конкретной формулы выходной последовательности.

2.2. Описание процесса обучения и оценки точности модели

Сам процесс обучения состоит из 11 итераций(заданий), для каждой из которых введены свои критерии обучения. К примеру для первой задаются такие параметры:

1. Learning rate = 0.001

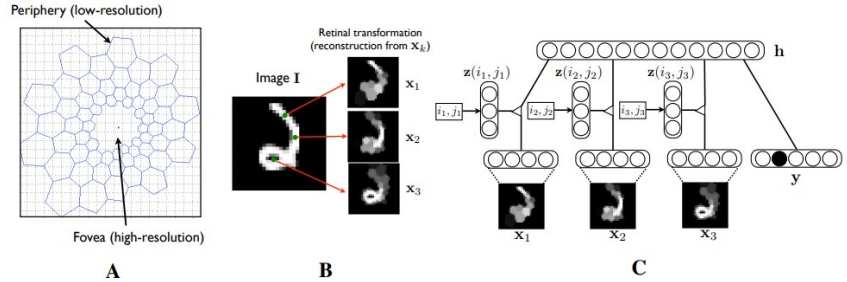


Рис. 2.1. Схема одного из первых примеров применения механизмов внимания в глубоких нейронных сетях при помощи машин больцмана терьего порядка, созданая Юго Ларошелем и Хинтоном[6].

2. Number of epochs = 100
3. Batch size = 20
4. ENC_DROPOUT = 0.5
5. DEC_DROPOUT= 0.5
6. DownsampleImage = False
7. Height of image = 60
8. Width of image = 360

Так как это самая первая итерация, тут нейронная сеть обучается 100 эпох на минимальном наборе изображений (1200 элементов). Но всё дальнейшее обучение будет проходить на большом "hugedataset"наборе изображений (104 000 элементов).
После прохождения одной полной итерации, мы получаем оценку работы нейронной сети, в частности расчёт её ошибки и BLEU (Bi-Lingual Evaluation Understudy) значение, которое показывает насколько распознанная нейронной сетью формула совпадает с правильной фомулой.

Validate average loss: 0.4935080707032117
BLEU score = 77.71

По мере прохождения обучения параметры нейронной сети будут изменяться следующим образом. После первой итерации будет изменён пакет данных с "generatedset"на "hugedataset". Далее для третьей итерации будет изменено значение "Learning rate":

Learning rate = 0.01

Для четвёртой итерации снова будет изменён параметр "Learning rate":

Learning rate = 0.0001

Для пятой обновлено значение "Batch size":

Batch size = 16

Перед запуском шестой итерации изменяем значение "DownsampleImage"с "False"на "True оно останется в этом положении до окончания обучения. Для запуска седьмой, поменяем сразу три параметра:

1. Learning rate = 0.00001
2. ENC_DROPOUT = 0.1
3. DEC_DROPOUT = 0.1

Для восьмой выставим исходное значение переменных "DropOut":

1. ENC_DROPOUT = 0.5
2. DEC_DROPOUT = 0.5

Для девятой сделаем такие изменения и переведём изображения в новый формат:

1. Learning rate = 0.0001
2. Batch size = 6
3. Height of image = 128
4. Width of image = 512

Для десятой внесём такие значения:

1. Learning rate = 0.00001
2. ENC_DROPOUT = 0.1
3. DEC_DROPOUT = 0.1

И, для завершающей обучение, одиннадцатой итерации изменяем эти параметры:

1. Batch size = 3
2. ENC_DROPOUT = 0.0
3. DEC_DROPOUT = 0.0

2.3. Результаты численных экспериментов

Итоги первой итерации:
Validate average loss: 0.4935080707032117
BLEU score = 77.71

Итоги второй итерации:
Validate average loss: 0.17141498625278473
BLEU score = 93.28

Итоги третьей итерации:
Validate average loss: 2.70721435546875
BLEU score = 17.54

Итоги четвёртой итерации:
Validate average loss: 2.249237537384033
BLEU score = 25.63

Итоги пятой итерации:
Validate average loss: 2.248100757598877
BLEU score = 26.14

Итоги шестой итерации:
Validate average loss: 2.2580454349517822
BLEU score = 24.55

Итоги седьмой итерации:
Validate average loss: 2.2548861503601074
BLEU score = 24.29

Итоги восьмой итерации:
Validate average loss: 2.4864206314086914
BLEU score = 16.67

Итоги девятой итерации:
Validate average loss: 2.5360090732574463
BLEU score = 14.42

Итоги десятой итерации:
Validate average loss: 2.2908096313476562
BLEU score = 21.89

Итоги одиннадцатой итерации:
Validate average loss: 2.1443867683410645
BLEU score = 25.05

Как можно заметить, значение ошибки и BLEU score после первых двух итераций сильно возросло, связано это с тем, что параметры обучения не изменялись, лишь был произведён переход с одного пакета данных на другой. Чего нельзя сказать после прохождения нейронной сетью 3 итерации, для прохождения которой был изменён параметр "Learning rate". Для четвёртой, пятой, шестой и седьмой мы получили примерно одинаковые значения, что уже говорит о более-менее стабильной работе нейронной сети при смене некоторых параметров. Перед седьмой итерацией мы изменили значение "DropOut что никак не повлияло, а когда вернули это значение в исходное положение перед восьмой итерацией, результаты сильно понизились. Для девятой так же наблюдается снижение результатов из-за нового формата изображений, но по итогу, в десятой и одиннадцатой мы наблюдаем улучшение результатов.

Заключение

В данной работе обучены две нейронных сети, одна из которых основана на GRU ячейке, а вторая на модели "Трансформер". Были получены две полностью рабочие нейронные сети, способные распознавать формулы любой сложности с изображений. По результатам обучения можно отметить, что нейронная сеть на модели "Трансформер"справилась с поставленной задачей намного лучше, чем нейронная сеть, основанная на GRU ячейке, даже с учётом того, что обе нейронные сети не всегда имели высшие показатели результатов в ходе обучения. Так же стоит отметить то, что сеть с GRU ячейкой, в отличии от второй сети, требовала в несколько раз больше времени и ресурсов для обучения, что не всегда является возможным для "домашнего"обучения.

Литература

[1] W.S. McCulloch, W. Pitts. A logical calculus of the ideas immanent in nervous activity // The Bulletin of Mathematical Biophysics. 1943.

[2] Bernard Widrow. Pattern Recognition and Adaptive Control // IEEE Transaction on Applications and Industry. 1964.

[3] Graham W. Taylor, Geoffrey E. Hinton, Sam T. Roweis. Two distributed-state models for generating high-dimensional time series // Journal of Machine Learning Research. 2011.

[4] Yuntian Deng, Sasha Rush, Hyliu. Neural model for converting Image-to-Markup // by Yuntian Deng zenodo.org/record/56198. 2016.

[5] Yuntian Deng, Anssi Kanervisto, Alexander M. Rush. A Visual Markup Decompiler // What You Get Is What You See. 16 Sep 2016.

[6] Larochelle H., Hinton G. E. Learning to Combine Foveal Glmpses with a Third-Order Boltzmann Machine // Advances in Neural Information Processind System 23. 2010.